Muhamaad Saad Amin

l253045@lhr.nu.edu.pk — +92 329 7930375 — LinkedIn — GitHub — Portfolio

Education

FAST NUCES (National University Of Computing and Emerging Sciences, Lahore
Bachelor of Software Engineering, Department of Computing *Aug 2025 – Aug 2029*

Unique Group of Institutions, Lahore
Intermediate Of Computer Science (ICS) *Aug 2023 – Aug 2025*

The Trust School (TEDDS), Lahore
Matriculation (Comp) *Aug 2021 – Aug 2023*

Experience

Full Stack Web Developer - MD Pharmacy , Lahore *June 2024 – Present*

- Developed A Full Stack **(MERN)** Website with complete backend (Node.js, Express.js, Morgan) and frontend (React.js) and integrated Database in **(MongoDB)**.
- Developed an Admin Panel for the pharmacy staff to update the products by just uploading the **Excel file** and the status of products between pending, shipped and delivered.
- Successfully deployed a full-stack web application on AWS using the **AWS Management Console** with optimized configuration for scalability and performance before I used vercel for deployment.
- Created **CSS (Cascading StyleSheet)** from scratch did not used any type of library such as Bootstrap and Tailwind CSS

MERN Developer - Pak Greenery SMC-Pvt Ltd *Oct 2025 – Present*

- Developed a robust PERN Stack application by integrating PostgreSQL, Express.js, React.js, and Node.js, ensuring smooth communication between frontend and backend while maintaining efficient API handling and database operations.
- Designed and implemented modern, responsive, and interactive frontend interfaces using Material-UI (MUI) components, customizing themes, layouts, and styles to enhance the visual appeal and user experience across different devices.
- Focused on optimizing UI/UX and frontend performance, applying best practices in state management, component structuring, and reusable design patterns to create a polished and professional application interface.

Freelance Web Developer - Fiverr *March 2021 – Present*

- Develop full-stack MERN web applications from scratch.
- Build custom admin dashboards for managing data (products, orders, etc.).
- Automate data updates via Excel/CSV file upload.
- Deploy and configure web apps on AWS (EC2, S3, Amplify) or Vercel.
- Implement user authentication, secure APIs, and cloud integrations.
- Add SMS or Email notifications using Twilio or Outlook API.
- Provide responsive UI/UX with animations and modern CSS effects.
- Offer bug fixing, optimization, and performance improvement for existing apps.

Projects

WEB DEVELOPMENT

MD Pharmacy *(Website)*
Developed a full-stack digital platform for MD Pharmacy in Pakistan, providing users with access to a comprehensive medicine catalog, prescription uploads, secure online ordering, and real-time notifications.

- **Frontend:** Built with React, HTML5, and CSS3 for a responsive, accessible, and interactive user interface; implemented reusable components and dynamic API-driven content.
- **Backend:** Developed scalable APIs using Node.js and Express.js with MongoDB integration; optimized search functionality and implemented real-time notifications via Pushbullet.
- Managed product lifecycle operations, including bulk updates through Excel integration, and ensured cross-browser compatibility and performance optimization.

Surface Lab *(Website) (Github)*
An interactive 3D mathematical surface visualization tool for plotting and analyzing $z = f(x, y)$ surfaces. Built with Three.js, WebGL, and Vanilla JavaScript, offering real-time rendering, advanced analysis, and export features.

- Real-time 3D surface plotting with support for multiple equations and color-coded visualization. Equation management with visibility toggling, highlighting, autocomplete, and function keyboard.
- Multiple rendering modes, color mappings, and customizable material properties. Analysis tools including critical points, surface normals, surface area, volume, and min/max values.
- Interactive camera controls, animations, and auto-rotate functionality. Export options including PNG, STL, CSV, and JSON for sharing and reuse.s Modular, scalable architecture optimized for performance and usability.

Boolforge *(Website) (Github)*
An interactive digital platform for designing, simulating, and analyzing Boolean logic circuits. Boolforge enables users to build custom circuits using drag-and-drop gates, visualize signal propagation, generate truth tables, and save/load designs for educational and professional purposes.

- **Interactive Circuit Design:** Drag-and-drop interface for creating logic circuits with AND, OR, NOT, XOR, NAND, NOR gates, and custom components.
- **Wire Connections and Signal Propagation:** Real-time wire connections between gates with dynamic signal visualization for accurate circuit simulation.
- **Truth Table Generation:** Automatic generation of truth tables for any circuit to analyze input-output behavior and validate logic designs.
- **Component Management:** Add, delete, and configure gates; highlight active components and toggle wire visibility for improved usability.
- **Save and Load Functionality:** Persist circuit designs locally or export for future use, enabling collaborative learning and project continuity.
- **User Experience Enhancements:** Undo/redo actions, keyboard shortcuts, and clean UI design for efficient and intuitive circuit creation.

GitHub Streak Viewer *(Website) (Github)*
A web application that visualizes GitHub contribution streaks and activity statistics using the GitHub GraphQL API, focused on clean data visualization, performance, and developer-centric UX.

- Built a responsive web app that fetches and visualizes GitHub contribution data using the GitHub GraphQL API with efficient query design.
- Implemented streak calculations, contribution summaries, and interactive charts to help developers track consistency and productivity.
- Optimized API usage and client-side rendering for fast load times and smooth user experience.
- Deployed a production-ready application with a clean UI, modular code structure, and scalable frontend architecture.

Canvas Market Frontend *(Website) (Github)*
React-based frontend for Canvas Market, a two-sided online marketplace connecting artists directly with buyers, collectors, and art enthusiasts to buy and sell original paintings and artwork, built with modern web technologies for a transparent, responsive, and user-friendly experience without traditional gallery intermediaries.

- **Responsive UI:** Mobile-first, fully responsive interface using CSS Modules with reusable custom layout, navigation, and feature components
- **Architecture and Routing:** Core pages (Home, Marketplace, etc.) with clear separation into components, pages, context, hooks, and utilities
- **Modular Design:** Scalable codebase using React Context API for state management (Redux planned)
- **Tech and Deployment:** Built with Create React App, React 18, ES6+ JavaScript, and deployed on Vercel
- **Future-Ready:** Infrastructure prepared for authentication, product listings, shopping cart, Stripe/PayPal payments, and advanced marketplace features

StoreFlow *(Website) (Github)*
StoreFlow is a full-stack order management system for shopkeepers, featuring a centralized dashboard, order creation, updating and deletion, customer-based search, purchase time tracking, and configurable settings for efficient store operations.

- Developed a complete e-commerce web application with React, Node.js, and MongoDB. Implemented dynamic product listing, category filtering, search functionality, and user authentication.

- Integrated shopping cart, checkout, and payment flow, ensuring real-time updates and a smooth user experience. Applied responsive design principles for optimal performance across devices.
- Built an admin dashboard for managing products, orders, and users, demonstrating full-stack development expertise.

Saad Web Experiments *(Website) (Github)*
Front-end project showcasing expertise in HTML and CSS through 54 web pages demonstrating practical implementation of web development concepts from basic structure to responsive design.

- Developed 54 web pages covering a wide range of HTML and CSS topics from FreeCodeCamp. Implemented responsive designs for multiple devices and screen sizes.
- Applied advanced CSS techniques including Flexbox, Grid, transitions, and animations. Used semantic HTML for accessible and well-structured pages. Created forms, tables, lists, media embedding, and interactive components.
- Built a personal portfolio of front-end experiments as a reference for HTML/CSS mastery.

PROGRAMMING

SFML Chess Game *(Website) (Github)*
A desktop-based chess game developed in C++ using SFML, showcasing strong object-oriented design, game logic implementation, and real-time graphical rendering.

- Developed a complete chess engine in modern C++ using SFML for graphics, input handling, and game-loop management.
- Implemented full chess rules including legal move validation, check, checkmate, and stalemate detection using efficient board-state algorithms.
- Designed an interactive graphical interface with piece sprites, move highlighting, mouse-based interaction, and menu navigation.
- Structured the project with modular components for rendering, game logic, input handling, and state management, demonstrating clean and maintainable C++ architecture.

Cryptocurrency Exchange Platform *(Website) (Github)*
A console-based Cryptocurrency Exchange Platform implemented in C++ using Object-Oriented Programming (OOP) principles. Simulates core exchange functionalities such as order books, wallets, CSV-based market data ingestion, and time-based trading simulation.

- **Order Book System:** Supports bid (buy) and ask (sell) orders Time-based order matching simulation Price statistics: minimum, maximum, and average
- **Wallet Management:** Insert and update currency balances Prevents invalid or negative transactions Exception handling for erroneous inputs
- **CSV Market Data Handling:** Parses historical market data from CSV files Converts rows into structured OrderBookEntry objects Maintains data integrity by ignoring malformed entries
- **Time-Step Simulation:** Market progresses in discrete timestamps Orders are evaluated per timestamp Enables realistic market simulation
- **OOP Design:** Clear class separation for modularity Encapsulation and clean header/source file structure Easy to extend and maintain

LeetCode Profile *(Leetcode)*
Competitive programming profile on LeetCode, the premier platform for algorithmic problem-solving and technical interview preparation. Completed about **400+ Problems** focuses on building proficiency in data structures, algorithms, and efficient coding practices using C++.

- **Problem Solving:** Solving algorithmic problems across multiple categories (arrays, strings, dynamic programming, trees, graphs, etc.) Emphasis on clean, efficient, and optimized C++ implementations
- **Technical Interview Preparation:** Practice with real-world coding challenges commonly used in FAANG-level interviews Development of systematic problem-solving approach and time/space complexity analysis
- **Continuous Skill Development:** Regular engagement with coding challenges to strengthen foundational computer science knowledge Preparation for competitive programming contests and software engineering roles
- **OOP and Language Proficiency:** Leveraging strong C++ fundamentals and object-oriented design principles from personal projects

Skills

- **Programming:** JavaScript, Python, C++, SQL
- **Frameworks & Libraries:** React.js, Next.js, Node.js, Express.js
- **Databases:** MongoDB, MySQL
- **Deployment:** Vercel, AWS Console
- **Tools:** Git, GitHub

Certificates

- **Introduction to Software Engineering** *Jan 2026 (IBM)*
- **Responsive Web Design** *Dec 2025 (freeCodeCamp)*
- **JavaScript** *Jan 2026 (freeCodeCamp)*
- **Python** *Jan 2026 (freeCodeCamp)*
- **Object Oriented Programming** *Jan 2026 (University of London)*
- **C++ Programming: Classes and Data** *Jan 2026 (University of London)*
- **Introduction to Object-Oriented Programming in C++** *Jan 2026 (University of London)*
- **Object-Oriented Programming in C++: Functions** *Jan 2026 (University of London)*
- **C++ Basic Structures: Vectors, Pointers, Strings, and Files** *Jan 2026 (Codio)*
- **Use C++ to build a Crypto Trading Platform: Final System** *Jan 2026 (University of London)*
- **Working with Objects in C++** *Jan 2026 (University of London)*
- **SFML C Game Development - Flappy Bird Clone** *Nov 2025 (Alison)*
- **18th Edition of NEO** *Oct 2025 (IEEE)*